CLAIMS

1. A computer implemented method for establishing communication between computing entities interconnected over a network, the method comprising:

(a) establishing communication with a web server to begin a communication exchange;

(b) executing a service proxy to manage the communication exchange;

(c) executing a dispatcher within the service proxy, the dispatcher configured to process the communication exchange, the processing of the communication exchange including,

(i) defining a request;

(ii) defining a container for enclosing the request;

(iii) defining a post request event for dispatching the container having the request; and

(iv) defining a response container for obtaining a response to the request; and

(d) transmitting the container enclosing the request and the response over the network to a responding entity;

wherein a response from the responding entity is received by the dispatcher in the response container.

2. The computer implemented method of claim 1, wherein the container for enclosing the request is a message class.

3. The computer implemented method of claim 2, wherein the post request event is configured to process the dispatching by,

(a) determining if a body is present for the container; and

(b) instantiating a post message class if the body is present, the post message class being configured to extend a get message class.

4. The computer implemented method of claim 3, further comprising:

a dispatchable interface for use by the post request event to process the dispatching.

5. The computer implemented method of claim 2, wherein the post request event is configured to process the dispatching by obtaining protocol handlers for a communication protocol of a URL of the responding entity at runtime to access local, remote and secure services in a common fashion.

6. The computer implemented method of claim 2, wherein the post request event is configured to process the dispatching by,

(a) determining if a body is present for the container; and

(b) instantiating a get message class if no body is present, the get message class configured to facilitate retrieval of data.

7. The computer implemented method of claim 6, further comprising:

a dispatchable interface for use by the post request event to process the dispatching.

8. The computer implemented method of claim 5, further comprising:

accessing a web server, the web server having an Internet protocol, a communication protocol, and a servlet.

9. The computer implemented method of claim 8, wherein the Internet protocol is TCP/IP, and wherein the communication protocol includes HTTP, HTTP/S, FTP, File, and Gopher.

10. The computer implemented method of claim 5, further comprising a service, wherein the communication between entities is between the service proxy and the service, and going through the web server.

11. The computer implemented method of claim 10 further comprising a delegate broker, wherein the delegate broker is part of the service and is in communication with a delegate registry.

12. The computer implemented method of claim 5, further comprising:

accessing the responding entity via file protocol directly from the dispatcher.

13. The computer implemented method of claim 5, further comprising a session ID, wherein the session ID enables stateful communication between the computing entities.

14. A method for enabling communication between a requesting entity and a responding entity, comprising:

receiving a request at the requesting entity;

obtaining the request by a service proxy, the service proxy configured to generate a message for the request;

dispatching the message by the service proxy to a service, the dispatching of the message being through a web server; and

receiving the message at the service having a delegate broker, the delegate broker being in communication with a delegate registry for associating the message with a delegate, the delegate being configured to communicate with the responding entity to process a transaction defined by the message.

15. The method of claim 14, further comprising:

obtaining protocol handlers for a URL of the service, the obtaining configured to enable the dispatching at runtime without compilation, and further configured to enable access to services including local, remote, and secure services.

16. The method of claim 15, wherein the protocol handlers include HTTP, HTTPS, FTP, and File protocol handlers.

17. The method of claim 16, further comprising:

instantiating a post request event, the post request event being instantiated in the service proxy and configured to:

(a) select between a PostMessage and a GetMessage;

(b) retain a session ID; and

(c) dispatch a Message;

wherein the Message is dispatched using a Dispatchable interface.

18. A computer readable media having program instructions for enabling communication between computing entities, the computer readable media comprising:

program instructions for invoking a service proxy to process a request by a requesting entity, the processing including:

(a) receiving the request from the requesting entity;

(b) generating a message for the request;

(c) dispatching the message to a service, the dispatching of the message being through a web server; and

(d) preparing to receive a response;

program instructions for receiving the message at the service, wherein the service includes a delegate broker;

program instructions for formatting a response, wherein the response is obtained by the delegate broker; and

program instructions for transmitting the response from the service to the service proxy.

19. The computer readable media of claim 18, further comprising:

program instructions for obtaining protocol handlers for a URL of the service, the obtaining configured to enable the dispatching at runtime without compilation, and

further configured to enable access to services including local, remote, and secure services.

20. The computer readable media of claim 19, further comprising:

program instructions for instantiating a dispatcher, the dispatcher being instantiated in the service proxy and configured to retain a session ID, the session ID configured to enable repeated communication between computing entities without instantiating a new dispatcher for each communication.